Click here to view PowerPoint presentation; Press Esc to exit

Dynamic VPNs for Coalitions

Dr. S. Zeber
Network Information Operations Section
DRDC Ottawa
3701 Carling Avenue
Ottawa, Ontario K1A 0Z4
CANADA

Steve.Zeber@drdc-rddc.gc.ca

J. Spagnolo
NRNS Incorporated
4043 Carling Avenue
Suite 106
Ottawa, Ontario K2K 2A3
CANADA

Joe.Spagnolo@nrns.ca

Dr. A.F. Gómez Skarmeta
Facultad de Informatica
Campus de Espinardo, s/n
30.071 Murcia
SPAIN

skarmeta@dif.um.es

Dr. G. Martinez Pérez
Facultad de Informatica
Campus de Espinardo, s/n
30.071 Murcia
SPAIN

gregorio@dif.um.es

ABSTRACT

Defence R&D Canada (DRDC) developed the dynamic virtual private network controller (DVC) prototype as a concept demonstrator for the rapid deployment and self-configuration of dynamic virtual private networks (VPNs) to support secure information exchange for dynamic multinational coalition operations, and has demonstrated the DVC prototype in both local and international environments. The establishment and management of the VPNs requires the coalition members to exchange configuration information and security access policies. When any of this information changes, the coalition VPNs must be reconfigured. Initially the configuration of VPNs required manual intervention by an operator. The DVC prototype is being extended to automate the configuration process by exploiting the concepts and technologies of policy-based network management (PBNM) systems. This paper describes the original DVC prototype, and the ongoing work to achieve a dynamic configuration capability using PBNM techniques, which is being done in collaboration with the Communications Research Centre (CRC) in Canada and the University of Murcia (UMU) in Spain. The paper also gives some guidance for the potential use of the DVC concept in a NATO environment.

1.0 INTRODUCTION

Modern military operations increasingly involve the deployment of multinational coalitions of combined joint forces whose operations require coordination among the coalition commander, the coalition land, sea, and air forces, and the national ministries of defence of the coalition members. Furthermore, the coalition membership is often dynamic and changes over time. This environment requires quickly deployable, easily managed, interoperable network communications, which provide the capability to exchange information securely in a timely manner among the various coalition members and force components in accordance with both coalition and national security policies. At the same time, both coalition and national information and network assets must be protected from unauthorized access by enforcing compartmentalization and need-to-know separation.

Zeber, S.; Spagnolo, J.; Gómez Skarmeta, A.F.; Martinez Pérez, G. (2006) Dynamic VPNs for Coalitions. In *Military Communications* (pp. 3-1 – 3-14). Meeting Proceedings RTO-MP-IST-054, Paper 3. Neuilly-sur-Seine, France: RTO.
Available from: http://www.rto.nato.int/abstracts.asp.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **DEC 2006**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Dynamic VPNs for Coalitions

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Network Information Operations Section DRDC Ottawa 3701 Carling Avenue Ottawa, Ontario K1A 0Z4 CANADA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM202750. RTO-MP-IST-054 Military Communications (Les communications militaires), The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**33**	

VPN technology, based on the Internet Protocol (IP) Security (IPsec) standards [1], shows promise in providing this capability as it enables secure communication over untrusted or unprotected network links. VPNs can provide authentication, privacy, and integrity for the information exchanged through these secured virtual communications paths. However, the technology has, so far, evolved largely to support static network configurations, and the process of configuring VPN devices is still mostly manual and error prone, since IPsec requires a great deal of configuration information that must be applied consistently to each VPN end-point.

NATO has been investigating the use of IPsec and VPN technology to support coalition operations through a multinational research project, Interoperable Networks for Secure Communications (INSC), that demonstrated a secure communications network infrastructure capability using the Internet IPsec technology and virtual private networks (VPNs) to support secure information sharing in a typical multi-national coalition operation. The results of the INSC project were presented at a symposium held at the NATO C^3 Agency in The Hague, in November 2003 [2], and have also been published in a report [3].

2.0 THE DVC CONCEPT

DRDC conceived and developed the DVC concept to support secure information exchange in a dynamic coalition environment. The DVC is a VPN device through which any coalition site can provide a remote coalition site with access to a protected network infrastructure and services, over a VPN connection , subject to an access security policy, without requiring any knowledge of the remote partner's protected network infrastructure. The DVC provides this capability by establishing a fully meshed network of point-to-point VPN connections among a set of coalition member sites that are assumed to operate at the same security level (Figure 1). Each point-to-point VPN connection encrypts the traffic between two coalition sites. Traffic sent over a VPN connection between two coalition sites is never routed through a third coalition site. The dynamic nature of the VPN is reflected in the fact that each DVC can establish, modify, and terminate VPN connections to the other coalition sites independently, at any time.

It is assumed that the coalition site DVCs are interconnected by an unprotected wide area IP network, such as the Internet. Each DVC identifies all remote coalition sites using the minimum information needed to identify a network access point: a name, which is a fully qualified domain name (FQDN), and a network address (an IP address). The name is authenticated by its binding to a trusted X.509 certificate.

VPN connections are subject to access security policies. Each DVC maintains a local policy file that contains, for each remote coalition site, the FQDN, the IP address, and a security access policy, which could be dynamic. The security policy describes which local subnetworks the remote site may have access to, what local services are provided to that remote site, and what domain name service (DNS) name bindings the remote site needs to make use of the local services. An access policy for a particular remote site is activated only when required and when authorized by a local security officer.

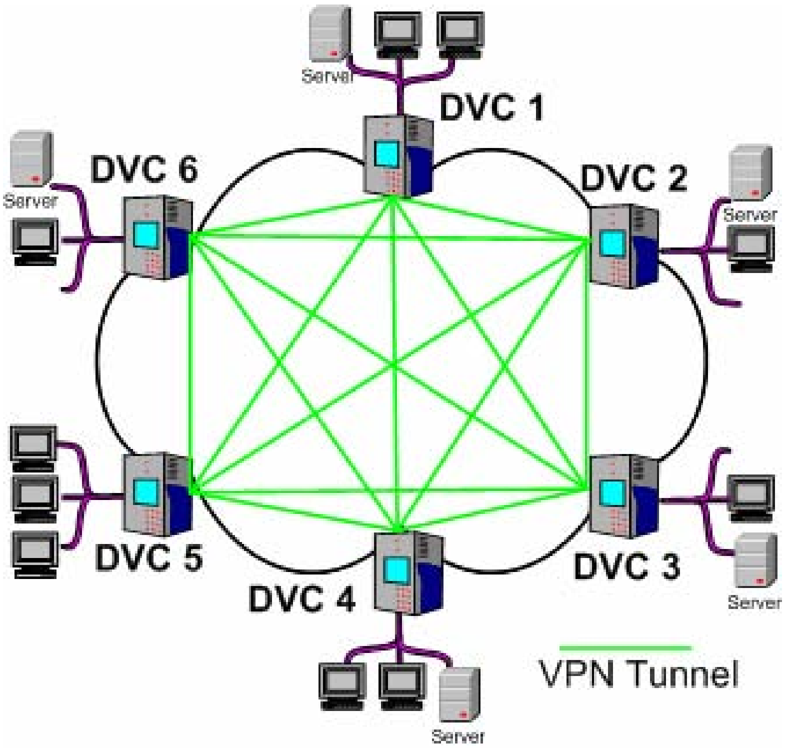


Figure 1: Fully-meshed VPN connections

To establish a VPN connection between two coalition sites, each site first authenticates the remote site's name, and then examines the access policy proposed by the other site. If the proposed policy of the remote site matches that configured in the local policy file then the local site accepts the proposed policy and authorizes the establishment of the VPN connection to the remote site. To complete the establishment of the VPN connection, both sites must modify the local configuration settings of their respective systems, including IPsec parameters, IP packet filters rules, routing tables, and DNS databases to match those configured for the remote site in the local policy file.

If a local site in an established VPN subsequently modifies its local access policies for a remote site, then the local site must notify the remote site and renegotiate the VPN connection with that site. If a local site in an established VPN terminates the access of a remote site, then the site terminating the access must notify the remote site and automatically terminate the VPN connection to that site.

3.0 THE DVC IMPLEMENTATION

Figure 2 shows the system architecture of the DVC prototype, which supports both IPv4 and IPv6 protocols. The system includes a main DVC process that executes all DVC functions, four subsystems to support the main DVC process: a Firewall subsystem, a Routing subsystem, a DNS subsystem, and an IPsec subsystem, plus a web-based Management Console and a Policy Editor.

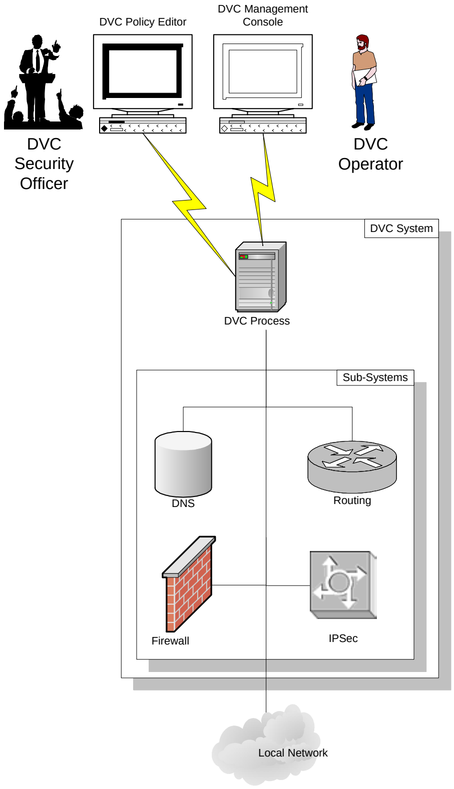


Figure 2: DVC System Architecture

The main DVC process implements all of the DVC functions. The process, which is written in Perl, is an event driven process that manages the subsystems, interacts with the Management Console and the Policy Editor and handles communications with remote DVCs. The Firewall subsystem ensures that only traffic configured in the policy is allowed across the VPN. The Routing subsystem advertises the remote networks accessible via the VPN. The DNS subsystem holds the name bindings required to access the services offered by remote coalition sites. The IPsec subsystem establishes and terminates VPN connections

The Management Console is a web-based application that presents a graphical view of the entire coalition membership and the status of all VPN connections. Using the Management Console, a DVC operator can establish and dismantle VPN connections to individual sites and to entire coalitions, manage the VPN connections for individual sites, and display the status of all coalition VPNs, from both the local perspective and the perspective of all other coalition members.

The Policy Editor enables a DVC security officer to create and manage local security policies that apply to the DVC. The Policy Editor is a Java-based application that facilitates the compilation of security policies encoded in the eXtensible Markup Language (XML) that conform to a defined schema. Objects representing local resources such as networks, name spaces (domains), services, servers, and permitted traffic are defined once and subsequently referenced by site level policies to eliminate the problems and potential errors resulting from erroneous data entry.

The DVC uses Secure Socket Layer (SSL) sessions, authenticated with X.509 certificates, as secure out-of-band control channels to negotiate, establish, and dismantle VPN connections, exchange policy information with remote sites, and monitor the health of the coalition VPN.

Figure 3 illustrates a typical DVC deployment in a two-member coalition. The DVC prototype has been demonstrated locally over the DRDC Research Network, in a six-node coalition scenario, and internationally, over the Internet, with nodes at DRDC, University College London (UCL) and the University of Murcia (UMU), in Spain. UCL has also demonstrated the DVC prototype over the 6NET.

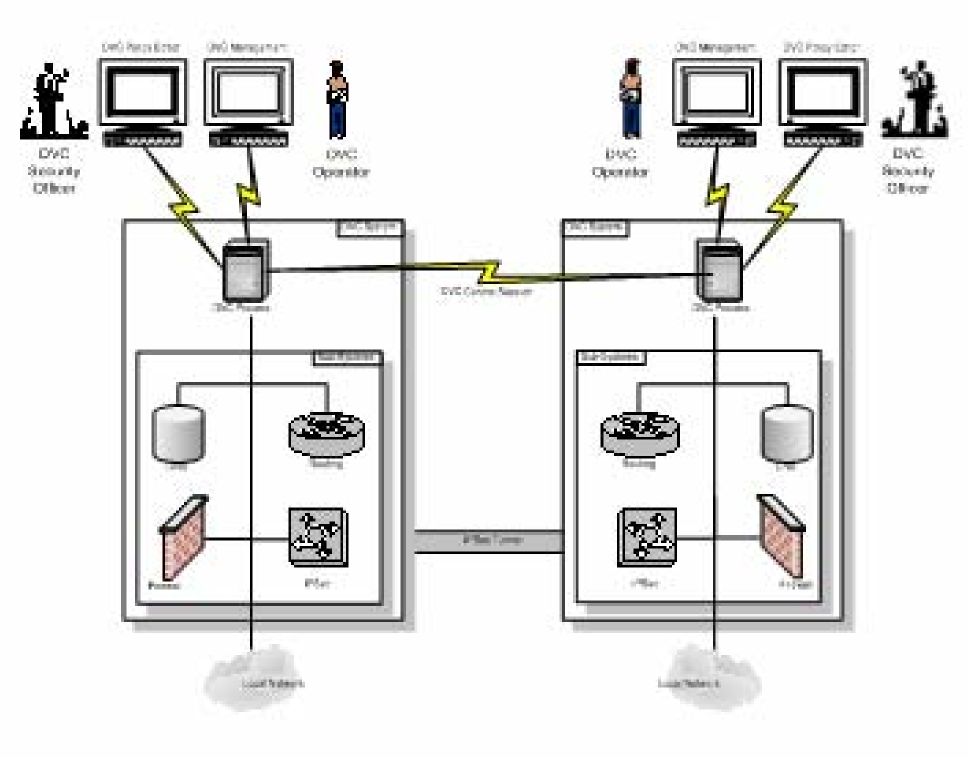


Figure 3: Typical DVC deployment for a two member coalition

4.0 SECURITY POLICY MANAGEMENT

A key feature of the DVC is the ability to specify and enforce different security policies for each VPN connection. Security policies are created and managed through the Policy Editor. These policies, best described as “inter-domain” security policies, define, for each remote coalition site, the local networks that require access to that remote site’s services over the VPN, which local services can be exported to the remote coalition site, and which DNS name bindings are needed by the remote coalition member site to make use of the offered services. The policies also identify which services the remote coalition site must provide and which services that site must not provide to the local site. The services exported to a remote partner’s site may depend on the identity of the particular partner.

The site security officer uses the DVC Policy Editor to create and install the security policy. After the DVC security policy is entered, it is compiled, and then must be pushed to a local DVC Policy Enforcement Point (PEP), which in the first prototype is the DVC itself. The compilation process converts the policy from a specification format used by the Policy Editor to a configuration format used by the DVC. Both formats use XML encoding. Finally, the policy is transmitted to the DVC over a secure, mutually authenticated SSL channel.

The DVC Policy Editor application is not co-located with the DVC system itself, but rather runs on a separate system residing in the private network protected by the DVC. The Policy Editor serves as the policy management centre, while the DVC serves as the policy negotiation and enforcement point.

The static configuration information required for IPsec to establish a VPN connection is derived when the DVC merges locally the configured security policy with the security policy proposed by a remote partner, after both partners have approved the policies. The DVC uses the merged policies to create the configuration information required by the co-located PEP, which not only enforces the negotiated policies but also facilitates the use of the VPN by manipulating both the local routing domain and naming services. This self-configuration reduces the management overhead and labour-intensive aspects of maintaining coherent VPN infrastructures.

5.0 EVALUATION OF THE FIRST GENERATION DVC

Although the first generation DVC prototype has been successful in demonstrating a technology for the rapid deployment of secure networks for a dynamic coalition, the demonstrations have also highlighted a number of limitations with the first prototype and indicated possible directions for improvements.

- Local security policies are stored in disk files as there is no central repository for these policies.
- The inter-domain security policies are simplistic and do not conform to any specific standard.
- The co-resident implementation of the policy negotiation and enforcement points inhibits use in a distributed environment and exposes the entire system on the unprotected network.
- When policies change and must be renegotiated, the DVC dismantles the existing VPN which results in a disruption of service.
- There is also no automated discovery mechanism, as the network identities of all potential coalition partners must be specified within the security policies. Ultimately, once a DVC is connected to the network, it should be able announce its presence, discover other coalition devices on the network, negotiate mutually acceptable policies and establish the desired VPN connections, automatically.
- There is no long-term archiving capability for policy documents as well as the artifacts produced during the negotiation process.
- Although the DVC uses SSL to authenticate and encrypt its negotiation dialogue, it does not offer any persistent authenticity and integrity capabilities for its policy documents or generated artifacts.
- The management of a fully meshed VPN does not scale well, as the performance degrades significantly and the Management Console displays become unusable when the coalition membership grows beyond ten members.
- The current implementation of the main DVC process and the four subsystems on the same physical system also leads to scaling, performance, and management problems that could be alleviated by a more distributed implementation.

6.0 LEVERAGING POLICY-BASED NETWORK MANAGEMENT

Policy Based Network Management (PBNM) systems provide an automated means to configure and administer policy enforcement point (PEP) devices such as virtual private network (VPN) devices, firewalls and routers. A policy decision point (PDP) takes high level policies from a policy repository as input and produces lower level PEP-specific policies as output. The PBNM system may contain many competing policies. When evaluating policies, the PDP must identify and resolve conflicts within different policies as well as take into consideration external factors such as the time-of-day and the current threat level.

Researchers at the University of Murcia (UMU) have recently developed an experimental PBNM tool [4] that demonstrates many of the PBNM concepts. The UMU PBNM system implements a distributed architecture and makes use of standardized protocols such as the Simple Network Management Protocol (SNMP) and the Common Open Policy Service (COPS) that facilitate the dynamic configuration of network and security devices [5]. DRDC is now attempting to leverage this PBNM technology for the DVC, in collaboration with the Communications Research Centre (CRC) and the UMU, to develop a second generation DVC, with a distributed architecture, capable of much more sophisticated and automated policy negotiation and management.

The second generation DVC system will focus exclusively on inter-domain policy negotiation and will divest its responsibilities for configuring PEPs to the UMU PBNM system. The UMU PBNM system provides a framework for the management of policies in IP networks based on the use of IPsec and public key cryptography as a way to deal with the security concerns associated with today's networked environments. The UMU PBNM system adopts the IETF approach to network policies by attempting to define the basis for supporting policy control mechanisms in a multi-vendor networking environment.

These second-generation inter-domain security policies will include time-of-day constraints, conditional statements, and other information that will assist in the evaluation of these policies. They will be based on the concept of an SLA (Service Level Agreement), which defines a basic abstraction through which administrative domains taking part in a coalition network can understand each other's capabilities, negotiate service parameters, as well as manage and monitor to agreed levels. The enhanced inter-domain security policies will permit policy to be described using high-level terms instead of device-level configuration items.

7.0 DVC-PBNM INTEGRATED ARCHITECTURE

The second generation DVC prototype is based on the concept of negotiating the inter-domain security policies for dynamic VPNs. A Policy Negotiation Point (PNP), an extension to the PDP, is responsible for negotiating policies with external parties. Although the immediate requirement is to implement dynamic VPNs with external organizations, the concept can be extended to support different types of policies, such as a privacy policy.

The integrated DVC-PBNM system includes five components as shown in Figure 4: a PDP, a policy repository, a policy editor, one or more PNPs, and one or more PEPs. The PDP serves as the system's nerve centre and interacts with all of the other components. The PDP also manages the policy repository and the PEPs. Figure 4 also illustrates four types of PEP devices. The firewall device houses the firewall and VPN PEPs. A router PEP and a DNS PEP are also shown. The VPN PEP establishes and maintains IPsec tunnels to remote sites; the firewall PEP enforces the negotiated service access rules; the router PEP advertises routes for remote networks within the local routing domain; and the DNS PEP distributes name bindings for servers in remote domains.

The system uses XML to express its inter-domain security policies. An inter-domain security policy includes an XMLSignature based digital signature to ensure the authenticity and integrity of the policy. The policy repository is a native XML database based on the Apache Xindice database. A native XML database allows XML encoded policies to be stored in their native XML form without the need to map the policies to some other data structure. Apache Xindice supports XPATH as its query language to facilitate the addressing of parts within an XML document. It also supports XUpdate to provide flexible update facilities to modify data in XML documents.

Three of the five components reside in the protected network and have no requirement to communicate with systems in the uncontrolled network. The firewall PEP maintains the boundary between the protected and uncontrolled network while the VPN PEP communicates with remote VPN PEPs to establish and maintain VPN tunnels. The PNP component however must undertake a dialogue with remote PNP components. As such, a separate system located in a protected demilitarized zone (DMZ) houses the PNP.

PBNM system components make use of various communication protocols. The policy editor employs Transport Layer Security (TLS) sessions to communicate with the PDP. The PDP and the PEP rely on the COPS or SNMP protocols for communication and the PDP interacts with the PNP using TLS. The PNP also leverages TLS to secure the communication with external PNP components in other administrative domains.

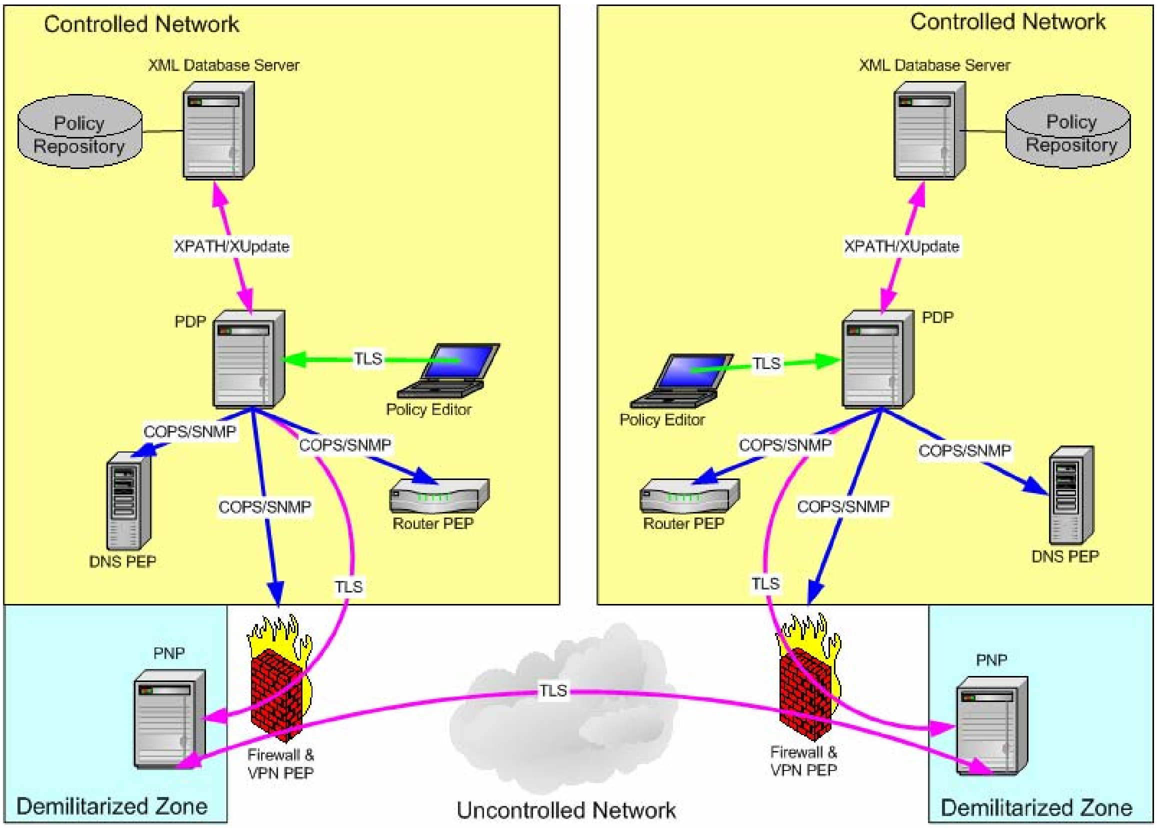


Figure 4: PBNM System Architecture

8.0 CONCEPT OF OPERATION

8.1 Introduction

The PBNM system merges locally defined policy objects with policy proposal objects from remote administrative domains (ADs) to implement dynamic VPNs. All local policy objects as well as all artifacts are encoded as XML documents that the PDP stores in their original format in the policy repository, thus preserving the digital signature. Specific artifacts include both local and remote policy proposal objects and the associated local and remote policy negotiation transcript objects. The PDP never overwrites a policy object in the policy repository. This allows old policies to be retrieved in support of auditing activities.

8.2 Policy Editing

The organization security officer uses the policy editor to create and update inter-domain security policies, which are stored in the policy repository and digitally signed by the PDP. The policy editor retrieves inter-domain security policy objects from the repository and submits policy objects to the repository over a secure TLS session with the PDP, mutually authenticated between the security officer and the PDP.

When a security officer retrieves a policy object for modification, the policy editor validates the authenticity and integrity of the retrieved policy object, using the PDP digital signature. After completing the modification, the security officer applies his digital signature to the policy object, and the policy editor then submits the policy object to the PDP. The PDP evaluates the updated policy and, if valid, adds a timestamp, the digital identity of the security officer and a unique policy identifier to the new policy. Finally, the PDP digitally signs the new policy object and saves it in the repository and proceeds to implement the new policy.

8.3 The Policy Negotiation Point

The PNP, as an extension to the PDP, undertakes the necessary dialogue with the remote PNP to exchange and negotiate policies. The PNP does not interpret or understand the policy negotiation objects that the PDP tasks it with handling. As such, the PNP must defer decisions regarding the evaluation, negotiation and acceptance of policies to the PDP.

The PNP uses TLS to communicate with each remote PNP. PNPs use their certificates to authenticate the underlying TLS session. Since PNPs interact as true peers, either PNP can initiate the TLS session. As such, the peer PNP devices detect race conditions that result in dual TLS sessions between them and eliminate the redundant session.

8.4 Policy Enforcement Point

The PDP merges the negotiated local and remote inter-domain security policies to produce a lower level policy suitable for PEP configuration. A negotiated inter-domain security policy results in lower level policy for up to four different type of PEP devices. The PEP devices employ the Common Object Open Policy for Policy Provisioning (COPS-PR) service to acquire its policies from the PDP.

COPS-aware PEP devices must be configured with the network address of one or more PDPs. When a PEP device boots, it establishes a COPS session to its primary PDP and supplies information to the PDP in the form of a COPS configuration request that describes the device’s capabilities (i.e. type of device, role). The PDP responds with all provisioned policies that are relevant to the PEP device. COPS structures its policy data as a tree-based namespace called a Policy Information Base (PIB), in which branches of the PIB represent classes, or types, of configuration data, and the leaves represent actual instances of those classes.

The PEP device maintains its COPS session active to the PDP at all times. This allows the PDP to push policy changes (additions, deletions) to the PEP when relevant policies are modified. The PEP device may transmit a new configuration request to the PDP at any time to report changes in its capabilities or simply to send status information to the PDP. The use of the persistent COPS session between the PDP and the PEP allows both devices to detect immediately when the other device reboots or fails.

The COPS protocol provides a mechanism to specify the encoding for objects encapsulated within COPS protocol data units. The COPS-PR designers identified a single protocol for this purpose - Basic Encoding Rules (BER). The University of Murcia extended COPS-PR in their JCOPS Java implementation to also carry XML encoded objects.

8.5 PDP/PNP Communication

The PDP uses TLS to solicit the PNP’s assistance with policy negotiation. Since the PNP can interact with numerous external PNPs concurrently, the PDP multiplexes the negotiation dialogue for all external PNPs over a single persistent TLS session.

8.6 Policy Negotiation Artifacts

The PBNM produces three types of artifacts as part of the negotiation process – policy proposal objects, negotiation transcript objects and merged policy objects. The PDP creates policy proposal objects for each individual AD. The PDP records its response to a policy proposal object in a negotiation transcript object. The PDP produces a merged policy object after a successful negotiation from the local and remote policy proposal objects. A merged policy proposal contains the details needed to produce the lower level PEP specific policies. The PDP applies a digital signature using XMLSignature to all policy negotiation artifact objects and stores them in the policy repository.

8.7 Policy Negotiation

The PDP signals the PNP to negotiate a policy with a remote AD and provides the local policy negotiation object. If the PDP discovers evidence of a previous successful negotiation with the remote AD for the same type of policy, the PDP instead supplies a signed policy refresh object. The policy refresh object identifies the most recent local and remote policy proposals.

PNP devices implement a full-duplex dialogue with each other. After they establish the session, the PNP devices may exchange policy refresh objects to determine if previously negotiated polices are still in force. If both systems are in agreement, negotiation is not required. If policy negotiation is required, the two PDPs, through their respective local PNP, exchange policy proposal objects and negotiation transcript objects.

If both PDPs accept the remote policy proposal (in whole or in part), each PDP merges the resulting local and remote inter-domain security policies to produce a merged policy object that it subsequently uses to configure the PEP devices. Each PDP also creates a new policy refresh object, which the PDP makes available to its local PNP.

8.8 Status Monitoring

After successful policy negotiation, the PDP engages the PNP in status monitoring of the remote PDP. Status monitoring involves the periodic exchange of policy refresh objects to determine if the negotiated policies are still force. When the remote PNP fails to deliver policy refresh objects within a specified interval, the PDP retrieves the associated merged policy object from the policy repository to retract the configuration previously supplied to the PEP devices.

A failure of a remote PDP to submit policy refresh objects can be attributed to network fault or a system crash involving the remote PDP or remote PNP. Regardless of the reason, the local PNP continues its status monitoring effort, which may require that it re-establish the underlying session. When the local PNP re-establishes communication with the remote PDP, the remote PNP may indicate that previously negotiated policies are still in force through a policy refresh object, or it may initiate a new policy negotiation dialogue if the remote AD modified its policies.

8.9 Policy Updates

When the security officer adds an additional remote AD to the inter-domain security policy, the PDP signals the PNP to initiate the negotiation with the new remote PNP. If the updated policy eliminates an active AD, the PDP signals the PNP to inform the remote PNP that the local AD will withdraw the previously negotiated policies on a permanent basis. The PDP also rolls back the associated device level policies within its PEP devices. At that point, the local PNP will no longer attempt to contact the remote PNP and will not accept sessions initiated by the remote PNP. The PNP only interacts with remote PNPs when permitted by the local inter-domain security policy.

If the updated policy alters the policy associated with an active AD, the PDP signals the PNP to re-initiate the negotiation with the remote PNP. Although the PDP leaves the previously negotiated policies in place until the PNP completes the renegotiation, the PDP establishes a time limit for the renegotiation to complete. If successful renegotiation does not occur within the allotted time period, the PDP signals the PNP to abort the renegotiation and inform the remote PNP that local AD will withdraw the previously negotiated policies. The PDP also rolls back the associated device level policies within its PEP devices. If the renegotiation is successful, the PDP makes only the required changes to the device level policies within PEP devices. This should minimize any disruptions associated with policy updates.

8.10 System Restart

The PDP stores all policy objects to persistent storage in the policy repository. This includes inter-domain security policy objects, local and remote policy proposal objects, local and remote negotiation transcript objects, as well as merged policy objects. If the PDP crashes and restarts, it can quickly re-establish the required VPNs without renegotiation with the use of a policy refresh object based on the policy objects stored in the policy repository.

9.0 SUMMARY

The DVC prototype is a promising technology that continues to evolve. The first prototype demonstrated a basic dynamic VPN capability with the negotiation of different security policies for each point-to-point connection. The integrated DVC-PBNM system provides an improved framework for policy negotiation. Its distributed architecture achieves more secure deployments since the exposed system, the PNP, yields no authority. The PDP, which resides in the protected network, signs all exchanged policy objects. The integrated system provides superior policy storage and audit capabilities since it never overwrites policy objects and policy objects always include a digital signature to preserve their authenticity and integrity.

The integrated system is better suited for operational environments since it minimizes the disruption associated with policy updates. Furthermore, it has the ability to recover quickly from system and network failures since it maintains a copy of all policy objects associated with the last successful policy negotiation. A simple exchange of policy refresh objects allows the PDP to implement the previously negotiated policies quickly. Finally the integrated system is also extensible to allow future integration of different policy types.

The DVC prototype as currently implemented would not be deployed for military use, principally because the use of open source software and software encryption on a commercial platform does not provide a readily assured security capability. However, the distributed modular architecture promises the possibility of developing a militarized version by integrating a military grade IP crypto module and by implementing other components on system platforms with an evaluated assurance level.

Even the integrated DVC-PBNM system does not yet achieved the full capability envisioned. A number of enhancements and extensions are contemplated, including the following:

- define the inter-domain security policies using standards such as the Common Information Model (CIM),

- introduce a higher level of abstraction for specifying communication requirements that would support the translation of a mission commander's mission communication requirements into inter-domain security policies,
- integrate military grade crypto devices and procedures,
- develop an auto-discovery and secure multicast capability, and
- develop the capability to monitor negotiated services.

10.0 CONCLUSIONS

The DVC has successfully demonstrated a prototype solution for the rapid deployment and self-configuration of VPN connections that provides a secure information exchange capability in a dynamic coalition environment. The solution provides interoperability based on IPsec, a standardized network security protocol, and it operates over both IPv4 and IPv6 networks. Because security is provided at the network layer, the solution is application independent and can be used to secure any application services that operate over an IP network. Furthermore, the use of policy negotiation for each point-to-point VPN connection provides the capability to provide different application services securely according to different security policies over each VPN connection. Finally, a deployable military capability for secure communications in a dynamic coalition environment appears realizable through judicious enhancements and extensions to the current prototype DVC.

11.0 REFERENCES

[1] S. Kent, R. Atkinson, *Security Architecture for the Internet Protocol*, IETF Request for Comments (RFC) 2401, November 1998

[2] Interoperable Networks for Secure Communications Symposium, NATO C^3 Agency, The Hague, November 4-6, 2003

[3] Interoperable Networks for Secure Communications Final Report, INSC/Task1/D011, March 9, 2004

[4] http://pbnm.umu.euro6ix.org/

[5] F. J. García Clemente, A. F.Gómez Skarmeta, G. López Millán, G. Martínez Pérez, *Secure VPNs over IPv6 Networks: An Evaluation and its Integration in a Policy Management Framework*, ISSN 1607-9264 Journal of Internet Technology, Vol. 5, n. 2 , 2004, pp.131-138.

12.0 GLOSSARY

AD	Administrative Domain
CIM	Common Information Model
COPS	Common Open Policy Service
CRC	Communications Research Centre
DMZ	Demilitarized Zone
DNS	Domain Name Service
DRDC	Defence R&D Canada
DVC	Dynamic VPN Controller
FQDN	Fully Qualified Domain Name
IETF	Internet Engineering Task Force
INSC	Interoperable Networks for Secure Communication, a research project conducted by eight NATO nations and the NATO C^3 Agency under a Memorandum of Understanding
IP	Internet Protocol
IPsec	Internet Protocol Security, an architecture and protocol standards
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
PBNM	Policy Based Network Management
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIB	Policy Information Base
PNP	Policy Negotiation Point
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SSL	Secure Socket Layer,
TLS	Transport Layer Security
UCL	University College London
UMU	University of Murcia, Spain
VPN	Virtual Private Network
XML	eXtensible Markup Language

DEFENCE R&D DÉFENSE

Dynamic VPNs for Coalitions

**Dr. S. Zeber (DRDC), J. Spagnolo (NRNS Incorporated),
Dr. G. Martinez Perez (UMU), Dr. A. F. Gomez Skarmeta (UMU)**

NATO RTO Symposium on Military Communications

IST-054/RSY-015

April 18, 2005

Defence R&D
Canada

R et D pour la défense
Canada

Canada

Outline

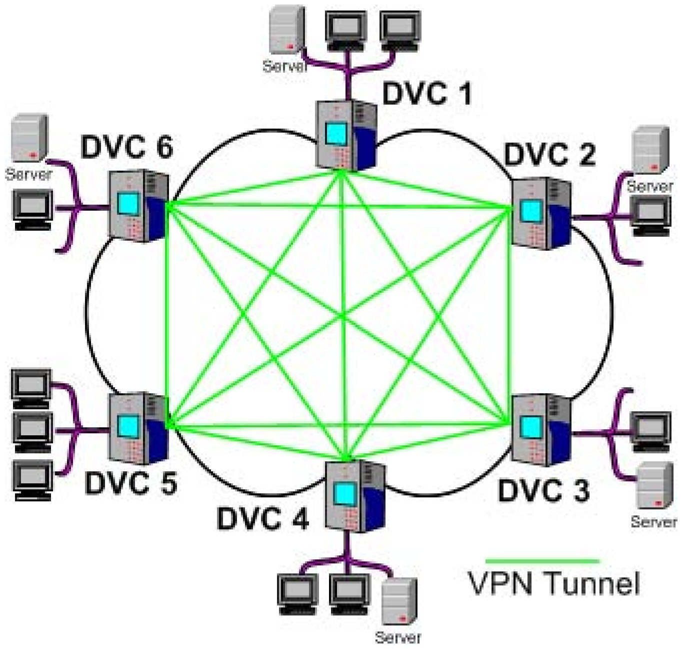
- Motivation
- DVC Concept & Architecture
- Principles of Operation
- Security Policy Management
- Policy-base Network Management (PBNM)
- Integrated DVC-UMU PBNM Architecture
- Concept of Operation
- Analysis and Conclusions

Motivation

- Most military operations involve coalitions
- Coalition membership and network can be dynamic
- Coalition communications requirements:
  - quickly deployable
  - easily managed
  - interoperable
  - secure

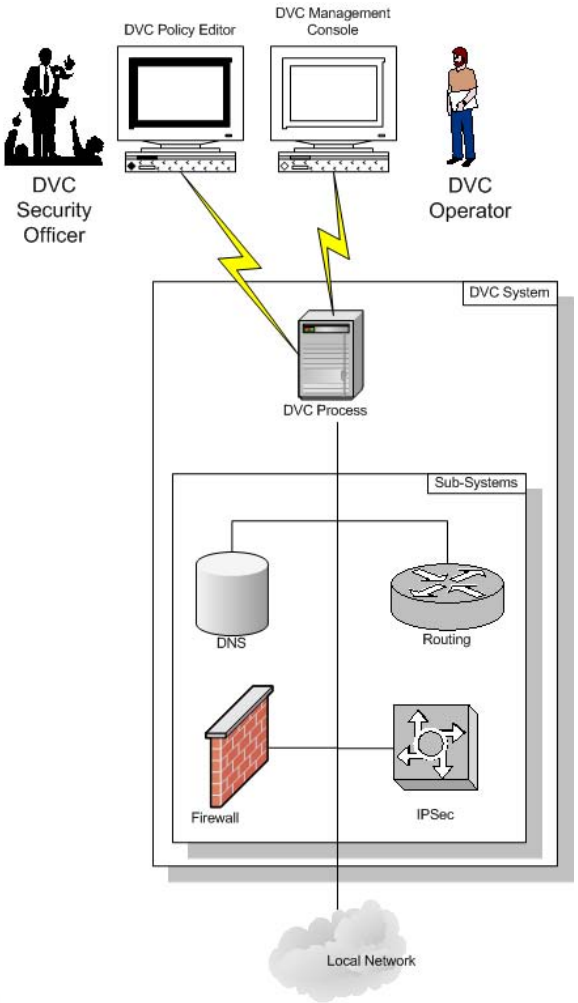
How do we rapidly deploy secure communications in this environment?

The DVC Concept



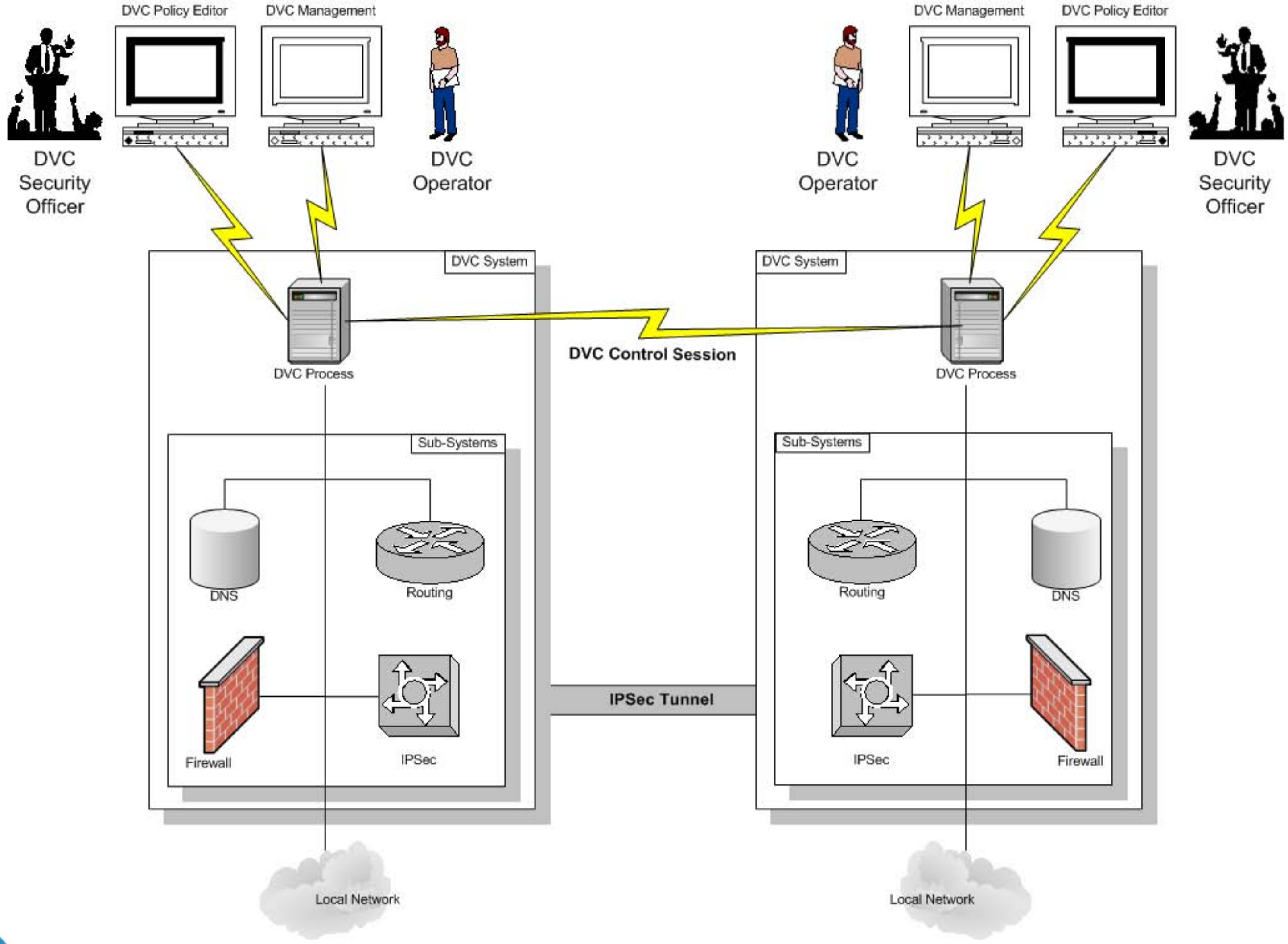
- The *Dynamic VPN Controller* (DVC) is a VPN boundary protection device
- Connects a coalition partner’s local network assets to the coalition WAN
- Establishes fully-meshed network of point-to-point VPN connections
- Each VPN connection is governed by a mutually negotiated and agreed security policy
- Each DVC monitors and reports the status of its VPN connections to the coalition
- A DVC can join or leave the coalition at any time

DVC Architecture



- Central DVC process
- DNS subsystem
- Routing subsystem
- IPsec subsystem
- Firewall subsystem
- Policy Editor
- Management Console

DVC for a Two-Member Coalition



Security Policy Management

- Each DVC maintains a local security policy database (SPD)
- SPD created and managed using the DVC Policy Editor and XML encoding
- SPD policies specify for each remote coalition site:
  - local services to the remote site
  - remote services imported from the remote site
  - DNS name bindings required
- Policies are exchanged, agreed and merged for each VPN connection
- DVC uses merged policy to configure local policy enforcement point (PEP) and subsystems

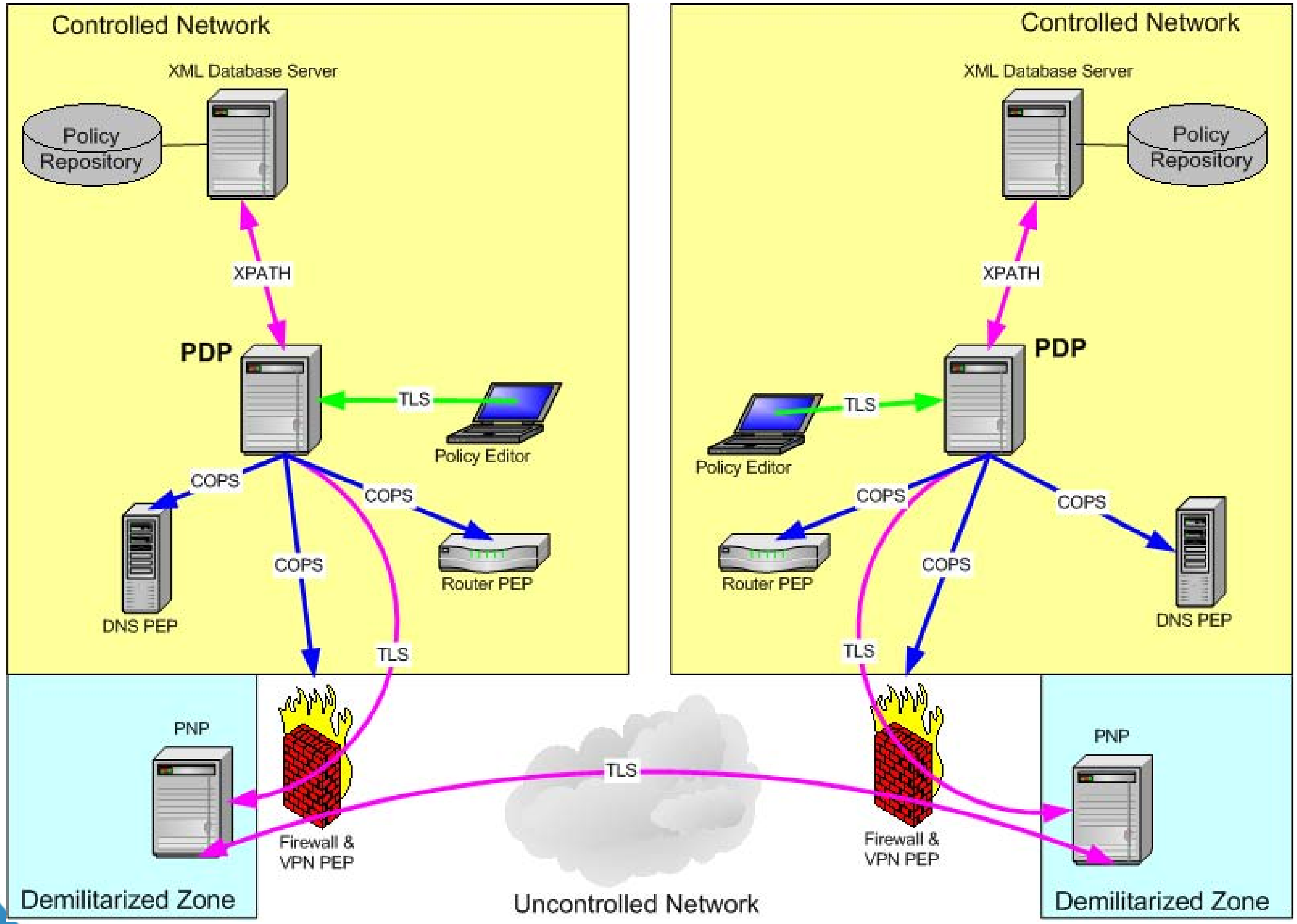
First Generation DVC

- Policies too simplistic to achieve true negotiation
- Simplistic policy management, lack of security, no policy archiving capability
- Co-resident policy negotiation and enforcement points
- No automated discovery
- Scaling, performance, management and security issues due to centralized implementation

Policy Based Network Management (PBNM)

- Leverage PBNM to improve DVC concept
- Based in part on UMU-PBNM implementation
- The concept of policy negotiation inherited from DVC
- *Policy Decision Point* (PDP) embodies the authorization function
- *Policy Repository* stores validated/authenticated Policy and Policy Negotiation artifacts – Policy Objects
- *Policy Negotiation Point* (PNP) facilitates policy negotiation
- Common Object Policy Service (COPS)-aware *Policy Enforcement Points* (PEP)

Integrated DVC-PBNM Architecture



Policy Objects

- Unique identifier
- Signed by the Policy Decision Point (PDP)
- Structure/Content not known to the Policy Negotiation Point (PNP) - opaque
- Stored in the Policy Repository
- Stored Policy Objects provide audit capability
- Stored Policy Objects allow policies to be implemented immediately at system start up through the exchange of Policy Refresh objects

Policy Negotiation

- Requires the exchange of Policy Proposal (PP), Negotiation Transcript (NT), and Policy Refresh (PR) objects
- A received PP is evaluated against local policy to produce an NT
- An NT is a statement by statement response to a received PP
- A PP can be accepted in whole or in part or can be completely rejected – critical versus non critical
- A pair of “acceptable” NTs denotes a successful negotiation and results in the creation of a Merged Policy object
- PR objects maintain negotiation state

Policy Enforcement

- PDP configures network devices based on policy – negotiated or static
- COPS for provisioning is used to push policies to network devices
- Network devices fulfill a “role” and are provided configuration items based on that role.

PBNM Framework

- An extensible PBNM implementation written in Java
- Framework can be extended to process different types of policies
- Currently Inter-Domain Security Policies (XML encoded) are supported
- The PBNM Framework (including the PNP) not concerned with encoding/structure/content of Policy objects
- Includes Java COPS (JCOPS) from University of Murcia, Spain

Inter-Domain Security Policy

- Three different nested policy scopes – Global, Coalition, and Administrative Domain (AD)
- Acceptance/Provision of services dictated by Local Policy Controls and Remote Policy Controls
- Policy statements inherited by inner scope from outer scope
- Policy Access Rules describe services that the local AD expects from the remote AD and services that the local AD is willing to provide to the remote AD – *What*, not *How*
- Each remote AD is assigned a priority
- Higher priority AD may cause reduction in service to lower priority AD

Future PBNM Work

- Support for additional policy types – both negotiated and non-negotiated (static)
- Integration with UMU-PKIv6
- External Event Processing
  - Change in threat level, Attack notifications
  - Network disruption notifications
  - Resource reservation requests
- Auto Discovery of PNP devices
- Common Information Model (CIM) compliant policies
- Audit Component to audit policy compliance

Conclusions

- DVC is a prototype for rapid deployment and self-configuration of secure coalition IPv4 and IPv6 networks
- Policy negotiation enhances dynamic capability
- PBNM Framework provides flexibility and extensibility
- PBNM Framework independent of data model
- Application-independent security based on IPsec
- Distributed PBNM architecture supports military hardening of selected components

Thank You

Questions?

DEFENCE R&D DÉFENSE